Hackers, software libre y anarquismo

Selección de textos

Este librito es *cultura libre*, debajo del título de cada artículo se encuentra la licencia que indica las condiciones de copia. En el artículo final «Definición de cultura libre» puedes encontrar la definición de *cultura libre*.

Código fuente de este librito

https://notabug.org/jorgesumle/hackers_software-libre_y_anarquismo

Editado por Jorge Maldonado Ventura (jorgesumle@freakspot.net)

Primera edición, 2018.

Índice

Copiar o no copiar, ¿he ahí el dilema?

Jesús M. González Barahona

Publicado originalmente en la revista TodoLinux
Número 23, pág. 12-13, Noviembre de 2002

Uno de los aspectos del software libre que más sorprenden al recién llegado es que el autor no sólo permita que su trabajo sea copiado y redistribuido libremente, sino que además anima al usuario a que lo haga. En general, tanto el autor como el usuario como quien recibe el programa que se copia quedan contentos. Y sin embargo, esto parece ir en contra de una idea que hemos oído muchísimo en los último años: *no se debe copiar software*. ¿Qué está ocurriendo aquí? ¿Habías pensado alguna vez sobre ello?

La máquina de duplicar pan

Olvidémonos por un momento de GNU/Linux, del software (libre o no) y de la copia de programas. Y fabulemos un poco. Imaginemos que, en alguna parte del mundo, en una prestigiosa universidad, alguien construye un invento completamente imprevisto: la máquina de duplicar pan. Es una máquina maravillosa. Casi no consume energía, la puede manejar cualquiera, se puede construir por millones a bajo coste, y no necesita materia prima ni mantenimiento. Su funcionamiento es simple: introduces una pieza de pan por un lado, y salen dos por el otro. La segunda es indistinguible de la primera: igual de sana, igual de nutritiva, con el mismo sabor. Y el proceso se puede repetir indefinidamente, usando las piezas de pan originales o las nuevas que ha producido la propia máquina. Muchos ya ven los problemas mundiales de hambre resueltos por fin. Se piensa inmediatamente en los duplicadores de lechugas, filetes, zanahorias, lenguados y otros muchos alimentos....

Pero antes de que nada de esto ocurra, comienza una campaña de publicidad en todos los medios. En ella aparecen drogadictos, asesinos, ladrones... y duplicadores de pan. El lema de la campaña es: *Cada vez que alguien utiliza el duplicador de pan, todos perdemos*. La campaña está dirigida por las asociaciones de panaderos, con dinero de toda la industria de la alimentación. Aparecen en todas las cadenas de televisión, en

todos los periódicos, en todas las emisoras de radio opiniones a favor de los panaderos y en contra de los que están empezando ya a duplicar pan. Se plantean cuestiones como *Si se permite la máquina de duplicar pan, ¿quién asegurará que tendremos innovación, y nuevos tipos de pan?* o *Si no se prohibe la máquina de pan, ¿de qué van a vivir los panaderos?* Inmediatamente se proponen legislaciones que prohiben la fabricación, comercialización y uso de máquinas duplicadoras de pan, y se empieza a considerar éticamente malo duplicar pan... Con el tiempo, las legislaciones se ponen en vigor, se crean ramas específicas de la policía para perseguir la copia ilegal de pan, y los panaderos, ya organizados, empiezan una campaña para que los productores independientes de pan tengan que pagar derechos por las recetas de los tipos de pan más habituales.

¿Una fábula sin sentido?

Bueno, volvamos a la realidad. ¿Por qué cuento todo esto? ¿Es que tengo una vena oculta de cuentacuentos con extrañas moralejas? No creo... Lo de las barras de pan se lo oí hace tiempo a Richard Stallman, y resulta que es algo que tenemos aquí ya... casi. Si en lugar de pan hablamos de programas, la máquina duplicadora la tenemos casi todos en nuestro PC. De hecho, tenemos varios tipos de ellas: Internet, disquetera, duplicador de CDs. Lo cierto es que desde hace unos años, los humanos disponemos de algo único en la historia: tenemos máquinas que pueden duplicar no sólo programas sino cualquier tipo de información a un coste prácticamente ridículo.

Hasta hace poco (en términos históricos) copiar información era caro y difícil. Los copistas medievales dedicaban su vida a ello, y sólo copiaban unos cuantos pergaminos a lo largo de ella. La imprenta mejoró mucho las cosas, pero no todo el mundo tenía una imprenta, y con ella no era económico hacer pocas copias. Y para distribuirlas y elegir qué se copiaba nació una inmensa industria: la industria editorial. La imprenta y esta industria, junto con otros factores, ayudaron a que la producción de información escrita creciera como nunca.

Cuando aparecieron los ordenadores y se empezaron a distribuir programas, se utilizó una organización similar a la del material impreso. Y así nacieron grandes empresas de software cuya labor es muy similar a la de los editores. Para que estas empresas pudieran funcionar, la sociedad, en todo el mundo, decidió que había que aplicar a los

programas una legislación similar a la de los libros, y en general se prohibió la copia de programas si el autor no daba permiso. Y el autor, normalmente, no daba permiso. En algunos países esto pareció poco, y se decidió que también se podía prohibir a los propios autores aplicar ciertas ideas a la hora de hacer programas. Las leyes que prohiben la copia de programas son las de derechos de autor (*copyright*) y las que prohiben la utilización de ciertas ideas son las de propiedad industrial (patentes). La motivación para estas leyes, en el caso del software, es similar: se supone que favorecen la innovación, aseguran que todos tengamos suficiente cantidad y calidad de programas, y permiten que los programadores vivan dignamente.

Pero todo esto no invalida el hecho fundamental: tenemos una máquina que permite duplicar los programas a coste prácticamente cero. Si la sociedad decide no usarla tiene que ser por poderosas razones. Si en algún momento quedase claro que puede producirse suficiente cantidad y calidad de programas sin prohibir la copia... ¿qué motivo tenemos para prohibirnos ese derecho?

Y, sin embargo, se mueve

Hace ya muchos años, en este océano de presión contra la copia de software, hubo un grupo de gente que nunca dejó de compartir sus programas, y de dejar que otros los repartieran a quien quisieran. Con el tiempo, esta comunidad creció y creció. De producir sólo unas cuantas herramientas para programadores pasó a generar miles y miles de programas para todos los públicos, desde procesadores de texto hasta navegadores de web. De estar compuesta fundamentalmente por voluntarios trabajando en su tiempo libre pasó a ser un hervidero de distintos tipos de gente, muchos pagados por empresas, muchos con sus propias empresas. De contar sólo con cantidades mínimas de dinero obtenidas vendiendo camisetas se pasó al dinero de las firmas de capital riesgo, de fondos de pensiones y de accionistas. De los individuos independientes y las empresas unipersonales se pasó a tener también multinacionales en el juego. Y de ser un puñado de desarrolladores en algunos lugares concretos se pasó a una comunidad de miles y miles de desarrolladores repartidos por todo el mundo.

Y a pesar de este proceso, que ha cambiado tantas cosas, que ha causado tantas tensiones, y que ha producido tantos programas, algo quedó siempre claro: si tú

recibes un programa libre, tienes derecho a copiarlo para tus amigos, para tus clientes, para quien sea. Y eso es bueno para ti, para tus amigos, para tus clientes... y para el autor del programa.

Es fácil entender cómo este mecanismo te beneficia a ti, a tus amigos y a tus clientes. Es más largo de entender cómo beneficia al autor del programa, pero es un hecho que es así. Muchos programadores viven ya del software libre, bien recibiendo ingresos directamente de quien lo usa, o bien cobrando un sueldo en alguna empresa que genera sus ingresos con un modelo de negocio basado en el software libre. Explicar cómo puede suceder esto, de dónde sale la financiación y cómo se puede crecer y ganar dinero si no se cobra por copia vendida es largo de explicar, y probablemente necesitaría otro articulo entero sólo para empezar. Pero a estas alturas es un hecho que esto ocurre, luego la pregunta ya no es tanto *¿Es posible?* sino *¿Cómo es posible?*.

Pero aún hay más. El modelo del software libre no sólo permite que tú copies y redistribuyas los programas que recibas. El modelo del software libre funciona mejor si lo haces. Cada vez que estás copiando un CD de GNU/Linux para un amigo, estás ayudando a que el software libre funcione mejor. Cada vez que un grupo de usuarios hace una tirada de CDs de Debian y los vende a bajo precio en una fiesta de instalación, está ayudando a que funcione el modelo del software libre. Cada vez que Red Hat, Mandrake o SuSE venden un CD en un hipermercado, están ayudando a que todos tengamos más y mejor software libre. De nuevo explicar esto es complicado, pero aquí sí es fácil sugerir ideas. Mayor número de usuarios supone un mayor mercado. Supone acercarse más a ser el número uno en ese nicho. Supone ser el estándar de referencia. Supone mucha gente interesada en aprender a usar ese programa, y mucha gente y muchas empresas dispuestas a pagar por servicios alrededor de ese programa. Muchos desarrolladores interesados en colaborar con mejoras y corrección de fallos. Cada vez que das una copia de un programa libre a un amigo, estás ayudando a que toda esta enorme rueda gire... en la dirección que más te beneficia.

La gran pregunta

Naturalmente, si todo esto es cierto (y hay millones de usuarios que dicen que sí es cierto), tenemos un modelo de producción de programas que ha demostrado que es capaz de producir suficiente cantidad y calidad para mucha gente. ¿Será capaz de generar suficiente calidad y cantidad para la mayoría de la gente? ¿Para toda la gente? Sólo el tiempo lo dirá, claro. Quizás todo esto no sea más que una burbuja que se desinfle en unos meses, y de la cual nadie se acuerde dentro de unos años. Quizás ninguna empresa sea capaz de encontrar un modelo de negocio que le permita tener ingresos saneados de forma estable. Quizás deje de innovarse en el software libre, y quizás nunca haya programas libres en muchos nichos. Pero si la tendencia actual continúa, la situación será más bien la contraria. Si seguimos por el camino de los últimos años, dentro de no mucho tiempo tendremos una saneada industria del software libre, con una poderosa comunidad de desarrolladores y usuarios satisfechos alrededor.

Y si todo es cierto, podemos volver a la gran pregunta, y decir: ¿qué motivo tenemos para renunciar al derecho a copiar programas? ¿Realmente es preciso prohibir la copia para que tengamos el software que necesitamos? Y más allá: si podemos tener el software que necesitamos sin prohibir la copia (ni de programas ni de ideas), ¿no sería mejor permitirla siempre, puesto que en ausencia de otros problemas los usuarios ganarían mucho?

¿Un cambio de tendencia?

Aún estamos dentro de una tendencia que parece llevarnos hacia más y más restricciones legales a nuestro derecho a copiar software. Las legislaciones sobre derechos de autor en informática son cada vez más estrictas, y las penas que se aplican son cada vez más grandes. Y quizás esto sea bueno para el desarrollo del software libre: cuanto más prisionero se encuentre un usuario de las empresas del software propietario, más motivado estará para probar con las opciones libres.

Pero en este entorno, es importante no perder de vista la situación de base: el único motivo para perseguir la copia es que eso sirva para motivar a los autores a desarrollar más y mejores programas. La única razón por la que en las sociedades democráticas podemos permitir que se nos obligue a pagar a un particular por algo que podríamos hacer gratis es porque eso beneficia a la sociedad en su conjunto (en el

caso del software, generando suficientes recursos para garantizar que se desarrolle más software de calidad). Si en algún momento esto dejarse de ser cierto, no habría muchos motivos para esta prohibición, ¿no crees?

Y al menos hay una comunidad (la del software libre) en la que esto ha dejado de ser cierto. Por ahora, aún no se ha demostrado el caso general, pero ya tenemos casos particulares. Así que atención a los próximos años... y ojo a las ideas preconcebidas. Si tienes un derecho, no renuncies a él sin buenos motivos. Sigue usando software libre y da una oportunidad a la realidad para cambiar... hacia mejor.

El software libre es mejor que la alquimia

Jorge Maldonado Ventura

¿Resulta difícil explicar las ventajas del software libre a personas que no entienden de ordenadores? Del mismo modo que no hay que ser periodista para entender los beneficios de la libertad de prensa, no hay que ser programadora para entender los beneficios del software libre.

El software libre garantiza cuatro libertades esenciales a las usuarias:

1. La libertad de ejecutar el programa como una desee, con cualquier propósito.
2. La libertad de estudiar el funcionamiento del programa y adaptarlo a las propias necesidades. El acceso al código fuente es un prerrequisito para esto.
3. La libertad de redistribuir copias para ayudar a las demás
4. La libertad de mejorar el programa y de publicar las mejoras, de modo que toda la comunidad se beneficie. El acceso al código fuente es un prerrequisito para esto.

1 https://creativecommons.org/publicdomain/zero/1.0/

Al garantizar estas libertades a las usuarias, el software libre reduce, si no elimina, toda posibilidad de dominación de las creadoras del programa sobre la usuaria. En el fondo se trata de impedir que nadie se aproveche del desconocimiento de las usuarias para tratar de controlarlas o aprovecharse de ellas de cualquier otra forma.

El concepto básico para entender el software libre es la libertad. Con el software libre las usuarias tienen la libertad y el control de hacer lo que les plazca con su computación, excepto privar de alguna de las cuatro libertades básicas a otra persona.

El software libre, por sus características, goza de ventajas que se pueden trasladar a muchos ámbitos:

- **Ecológico**. El software libre es completamente reutilizable, no es cómplice de la obsolescencia programada y del consumo descontrolado.
- **Ecónomico**. El software libre es muy barato, la mayoría de las veces gratuito, pues siempre se puede reutilizar. Normalmente se suele pagar por la programación de funcionalidades muy concretas o por servicio técnico.
- **Educativo**. Es imposible aprender cómo funciona realmente un programa si no se accede al código fuente.
- **Ético**. Responde a la reclama de que todo el conocimiento debe ser libre y no estar sujeto a ninguna forma de censura o restricción.
- **Político**. El control de la población es muy difícil con el software libre, pues, al ser accesible el código fuente, cualquier persona puede detectar las funcionalidades maliciosas.
- **Privado**. La privacidad es una mayor garantía con el software libre. El espionaje masivo mediante funcionalidades ocultas en programas no es posible, al ser el código auditable.
- ...

El software libre es empírico y sigue un método de trabajo científico. Por otro lado, el software privativo se basa en el oscurantismo, en la fe o en la buena voluntad o habilidad de sus creadoras. Básicamente, es como comparar los métodos de trabajo y las creencias de la alquimia con los de la ciencia.

El software libre es activismo

David Arroyo Menéndez

«No odies a los medios, conviértete en los medios». Este era el lema con el que surgieron los blogs de contrainformación en los '90 y principios de los 2000: periódicos en una red llamada Internet en la que cualquiera pasaba de ser consumidor a constructor de manera rápida, y proyectos como sindominio[3], ourproject[4], riseup[5], etc., lo ponían en práctica, aunque fue la red Indymedia la forma por excelencia de contar las cosas en una lucha global contra el capitalismo.

Sin embargo, ¿qué ocurre con Facebook y Twitter? **Ahora el capitalismo puede decir algo como «si ellos son los medios, que sean el producto»**. Y eso es exactamente lo que pasa: la gente retwitea, pone «me gusta» en Facebook y todo el mundo encantado. Hay revueltas en el mundo árabe que pasan a democracias, que como mucho se parecen en lo malo a las occidentales y en otros casos acaban en cruentas guerras civiles. Y en España hemos tenido 15M, que se infló y se desinfló como un globo especulativo más. Ahora le toca el turno a los sindicatos y a las huelgas, con los que puede pasar tres cuartos de lo mismo.

¿Todo esto ocurre por las redes sociales o por los brutales ataques del capitalismo a la clase trabajadora? **El problema es que con las redes sociales quienes las controlan especulan más y mejor**. Criticar a un partido se convierte en un ejercicio que sube y baja como la bolsa, mientras los expertos en minería de datos de quienes nos gobiernan (ricos, banqueros, políticos) hacen sus apuestas con las que cada vez el 1% es más rico y el 99% más pobre. ¿Cómo se sale del bucle? Rompiendo con el capitalismo y creando proyectos de manera horizontal, comunitaria y justa, al estilo de las colectividades anarquistas de los años 30.

Está claro que las reglas del juego han cambiado. Ahora disponemos de otro desarrollo tecnológico y cabe preguntarse qué hacer con él. Si para difundir el

2 https://www.diagonalperiodico.net/saberes/software-libre-es-activismo.html
3 https://wp.sindominio.net/
4 http://ourproject.org/
5 https://riseup.net/es

mensaje tenemos tecnología, podemos usarla como en los '90, pero siendo los medios, no los productos. En este sentido usar redes sociales libres y federativas (Diaspora*[6], Mastodon[7], Peertube[8]) es lo coherente para los tiempos que corren.

¿Cuál es el papel del software libre?

El software libre **se define como aquel software que otorga una serie de libertades a sus usuarios/as:** uso, copia, modificación y redistribución de lo modificado. De este modo, el software libre se puede entender como propiedad colectiva generada por sus usuarios/as y productoras/es.

Sin embargo, desde la aparición de la filosofía Open Source, esto ha venido cambiando paulatinamente hasta el punto de que si bien el software libre sigue siendo propiedad colectiva, cada vez menos es generada por los intereses reales de usuarios y productores, sino por intereses del mercado capitalista, con criterios de especulación y burbujas financieras dignas de cualquier otro producto bursátil. De este modo, **la filosofía Open Source expropia las argumentaciones éticas del discurso de si es razonable o no utilizar software libre**. El único argumento válido será si técnicamente es o no una buena opción, si es o no rentable y argumentos de este estilo. Pero no se plantea lo que es adecuado para el bien común, eso se sale del discurso.

Muchas empresas han hecho contribuciones al mundo del software libre desde ese discurso, creando productos mayoritariamente de software libre desde esa filosofía, pero muchas veces mezclada con la filosofía del software privativo: Ubuntu, Android, etc. En un mundo capitalista el triunfo de la rentabilidad y la tecnicidad es fuerte, por eso **finalmente se ha abandonado la argumentación original de crear software que no busque una rentabilidad económica, sino social**, es decir, crear software de una manera que sea útil para quien lo genera y lo usa. Aún quedan muchas comunidades como Debian, GNU, etc, que siguen fieles a la producción de software con utilidad social antes que a la rentabilidad económica cortoplacista.

6 https://diasporafoundation.org/
7 https://joinmastodon.org/
8 https://joinpeertube.org

No es fácil pensar a largo plazo y salirse del capitalismo en lo tecnológico o en cualquier otro aspecto, pero es lo correcto y es el objetivo para conseguir la felicidad colectiva y acabar con la desigualdad exacerbada actual.

Así que si alguien te pregunta «¿Qué herramientas tecnológicas has incorporado debido a tu activismo político?», por favor, responde «uso software libre basado en la comunidad, como GNU Emacs, Debian, Trisquel, Gimp, o GPG, porque **aspiro a que el mundo del software se organice de manera cooperativa** y justa».

¿Por qué el software privativo va en contra del espíritu educativo?

Jorge Maldonado Ventura

El software privativo no forma parte del conocimiento de la humanidad, porque ni siquiera se puede estudiar: permanece secreto y restringido.

La cooperación con el software privativo también está restringida. No puedes compartir sin incumplir la ley. Al no poder estudiar el código fuente, no se pueden realizar modificaciones ni averiguar cómo funciona.

La compañía de software o el particular es quien tiene el poder; no el usuario. En cualquier momento pueden dejar de desarrollar la tecnología, deshacerse de ella, subirle el precio, utilizarla para espiarte... En resumen, dejas de tener el control: estás en manos de la buena voluntad de una empresa.

Si nos centramos en el aprendizaje de programación, por ejemplo, observamos que con software privativo es prácticamente imposible. Los programas privativos impiden la colaboración de los usuarios y programadores en su diseño y desarrollo. Por lo tanto, no puedes hacer siquiera una pequeña modificación a un programa. No se aprende a programar realizando ejercicios simples y sin sentido, sino modificando software y colaborando con proyectos reales.

Las empresas de software privativo tienen mucho interés en hacer llegar sus productos a las escuelas para inculcar una dependencia hacia ellos, por eso muchas

veces los dejan gratis o a un precio muy reducido. No hacen un favor a los alumnos, al contrario. Cuando a un alumno le inculcan una dependencia hacia el software privativo, le están haciendo débil frente a corporaciones cuyo único fin es aumentar sus beneficios a cualquier precio. Los alumnos que quieran aplicar lo poco que han aprendido tendrán que gastar mucho dinero para poder utilizar productos a los que dichas empresas den soporte; aún así, nunca sabrán cómo funcionan.

La conclusión que podemos alcanzar es que el software privativo es completamente incompatible con la buena educación y con la libertad de los alumnos.

> *«Así, porque las escuelas deben enseñar el espíritu de buena voluntad, el hábito de ayudar a otros a tu alrededor, cada clase debería tener esta regla: estudiantes, si traéis software a clase, no podéis quedároslo para vosotros».* —Richard Stallman

Anarquía y el código fuente

¿Qué tiene que ver el movimiento de software libre con el anarquismo?

En un restaurante de Nueva York, dos hombres estaban almorzando y meditaban sobre los pasos a seguir para su pequeña revolución. Uno de ellos, Eben Moglen, reflexionaba sobre la impresión que le darían a la gente que pasaba a su lado: «Aquí estamos nosotros, dos anarquistas barbudos, reflexionando sobre los pasos a seguir. Cualquiera que escuchara nuestra conversación pensaría que estamos locos. Pero yo estoy seguro que sentados juntos en esta mesa, estamos preparando una revolución». La persona que estaba sentado al frente, Richard Stallman, sería el que realizaría una revolución (ver Williams 2002, p.184)

9 https://creativecommons.org/licenses/by-sa/2.0/de/

No sólo Eben Moglen, Profesor de derecho de la Columbia Law School, se considera un anarquista. Richard Stallman, uno de los personajes más destacados del movimiento de software libre, también se define así. No debemos imaginarnos a los anarquistas del software libre bajo el cliché de unos extremistas con cabello despeinado, mirada loca y una bomba en cada mano. Todo lo contrario, en vez del caos, Stallman reclama un nuevo orden de la propiedad intelectual visto desde una ética de hacker: el acceso al saber debe ser libre, descentralizado, antiburocrático y antiautoritario. Como norteamericano, Stallman no tiene ningún escrúpulo en denominarse anarquista ya que el término anarquista en EE.UU tiene un significado diferente que en Europa. Para David DeLeon, Profesor de Historia de la Universidad Howard en Washington D.C., el anarquismo es la única crítica constructiva radical a la sociedad liberal de EE.UU, como lo indica en su libro *The American as Anarchist*. Además, esta es mi hipótesis: el anarquismo norteamericano es una condición clave de la ética hacker (Imhorst 2004).

El mensaje de Stallman es un mensaje político radical ya que tiene como objetivo la propiedad privada, uno de los pilares de la sociedad en la que vivimos. La propiedad intelectual representada por el software es la forma de generar dinero del recién pasado siglo XX y del siglo XXI que acaba de comenzar. En fin de cuentas, Bill Gates, el hombre más rico del mundo no adquirió sus riquezas haciendo negocios con petróleo, oro o especulando con acciones sino vendiendo software. Desde 1980, a través de Copyrights y patentes sobre la propiedad intelectual en forma de software se puede ganar miles de milliones de dólares.

Los enemigos de Stallman le reprochan su interés en abolir la propiedad intelectual y el que busque realizar una utopía comunista con su movimiento de software libre. Él mismo no se considera comunista o un enemigo público anticapitalista que quiera suprimir la propiedad. La licencia de Stallman, la GNU General Public License (GPL), expresión del espíritu libre del movimiento de software libre, tampoco se refiere a la abolición de la propiedad intelectual. Al contrario, quiere preservar una forma específica de la propiedad intelectual.

A través de la GPL, el programador de software libre traspasa el control de su obra pero no así la obra misma. El se reserva el derecho como autor sobre su programa. Se le conceden ciertas libertades al usuario de este software, como la libertad de modificar la obra y de publicarla modificada. Esta libertad depende sólo de una condición: la obra modificada también debe ser publicada bajo la licencia de GPL. Ya existen licencias similares para libros, música y otras formas de propiedad intelectual.

Estas libertades no pueden ser suprimidas por nadie. El software libre no debe ser la propiedad de un individuo sino la propiedad de todos. Lo contrario es el software propietario.

Un programa propietario como Microsoft Word es propiedad privada de la empresa Microsoft. La persona que instala Word sólo tiene derecho de uso sobre el programa. La licencia extensa existe para proteger Word de ser repartido y modificado. Al contrario, la licencia GPL exige la modificación y la repartición del software. Nadie está excluído de la propiedad de un software de licencia GPL. Por esta razón, es imposible que alguien controle la distribución. La persona que quiere usar este software puede simplemente copiarlo y distribuirlo. De esa forma, aumenta rápidamente la disponibilidad del programa. Sin duda, la licencia GPL impide que alguien sea excluído del uso del software libre pero por otro lado, imposibilita que alguien haga de un software libre uno propietario. A nadie se le puede impedir que use el sistema operativo libre GNU/Linux y a nadie se le puede restringir su uso. A cualquiera que descarge a través del internet GNU/Linux, lo instale en su ordenador, regale o venda copias del mismo, también le pertenece. En este sentido, la licencia GPL es una anti-licencia, por lo que Stallman habla de Copyleft en vez de Copyright. La ética hacker es fundamental para la filosofía política de Stallman. Es un código que se atribuyó un grupo de aficionados en computación del Massachusetts Institute of Technology (MIT) a finales de los años cincuenta. Ellos aprendieron juntos a programar los primeros ordenadores en el MIT y compartieron estos conocimientos entre ellos. Denominaron hackear el programar juntos, aprender e intercambiar libremente los conocimientos y a ellos mismos como hacker, antes de ser denominados piratas cibernéticos por los periodistas. La ética hacker tiene sus elementos anarquistas en la exigencia de la libertad y la descentralización, así como en sus tendencias antiburocráticas y antiautoritarias.

Mientras que el anarquismo ha desaparecido mayoritariamente en Europa, sigue existiendo en la tradición norteamericana. En su libro *The American as Anarchist*, DeLeon explica este fenómeno a través de tres características esenciales del modo de vida norteamericano: el protestantismo radical como religión sumamente interiorizada, un gran territorio en el que las diferentes comunidades pudieron escapar al control del estado y el capitalismo anárquico norteamericano.

> Nuestros radicales se concentran en la emancipación, quieren romper las cadenas del poder en vez de forjar nuevas. Son salvadores y no fundadores de instituciones, reinvendican los derechos de la mujer, de los

> homosexuales, de los negros y luchan por una teología de la liberación; son profetas en vez de curas; anarquistas y no administradores. En general, suponen que el espíritu liberado necesita poco o ningún liderazgo. (DeLeon 1978, p.4)

Radicalismo norteamericano

Después de dos siglos de independencia, el anarquismo en los EE.UU. se diferencia fundamentalmente del anarquismo europeo o ruso. Los habitantes de los EE.UU. se dieron una propia identidad nacional cuando se emanciparon de Europa. De este modo, crearon su radicalismo liberal propio del «*new lands*, *new men, new thought*» (Nuevos territorios, nuevos hombres y nuevos pensamientos). El radicalismo norteamericano era algo nuevo y no solamente una variación del radicalismo europeo. Los anarquistas norteamericanos nunca quisieron suprimir toda autoridad. Eran representantes de una nueva forma de orden, el orden del anarquismo norteamericano. Lo problemático del concepto de anarquismo es que nunca puede ser una doctrina o una teoría estática. El anarquismo puede ser reformulado por cualquiera de sus seguidores y representado en formas diferentes. Esto tuvo como efecto en EE.UU. la división entre un anarquismo de «derecha» y uno de «izquierda». No se puede esperar que los anarquistas tengan una teoría política consistente ya que el anarquismo es una teoría de adaptación a las circunstancias. Es precisemente esa capacidad de adaptación que hizo que el anarquismo nos haya acompañado a través de la historia de la humanidad y que se expresó en los últimos tiempos intensivamente en el movimiento hippie.

Los hippies anarquistas de California fueron los pioneros de la contracultura de los años sesenta e influyeron los movimientos izquierdistas del mundo entero. Con su forma política de «acción directa» (un término más antiguo de la tradición anarquista es el de «la propaganda de la acción») organizaron campañas en contra del militarismo, del racismo, de la discriminación sexual, etc. En su ensayo «La Ideología Californiana», los sociólogos ingleses Richard Barbook y Andy Cameron los denominaron «liberales en el sentido social de la palabra». El movimiento hippie no creó jerarquías como la izquierda tradicional, sino colectivos y estructuras democráticas.

> Además, la izquierda californiana combinó la lucha política con una rebelión cultural. Al opuesto de sus padres, los hippies se negaron a vivir bajo las estrictas convenciones sociales a las cuales la gente tenía que someterse en el ejército, en las universidades, en las empresas y también en los partidos políticos de tendencia izquierdista. Por el contrario, mostraron su repudio en contra del mundo ordenado vistiendo ropa poco convencional, viviendo una promiscuidad sexual, escuchando música a todo volumen y tomando drogas relajantes (Barbrook y Cameron 1997).

El hacker anarquista

Los primeros aficionados de computación del MIT no se interesaban en la promiscuidad sexual ni en la música fuerte o las drogas relajantes. A pesar de eso, seguían teniendo algunas cosas en común con los hippies, es decir, el rechazo por las estructuras autoritarias y burocráticas y la exigencia de su superación. En su libro *Hackers – Heroes of the computer revolution*, Steven Levys dice que la subcultura del hacker se creó a finales de los años cincuenta en el MIT. Durante la primavera de 1959, la universidad impartió su primer curso de programación. Usar un ordenador significaba en aquel entonces algo que requería mucho esfuerzo. Los ordenadores eran máquinas gigantescas que se programaban a través de tarjetas perforadas y no tenían pantallas. Por lo tanto, antes de poder programar un ordenador con tarjetas perforadas, había que obtener el visto bueno de los ingenieros que supervisaban las computadoras y se autodenominaban sacerdotes. Cuando los primeros hackers como Peter Samson, Bob Saunders o Alan Kotok intentaron usar unos de esos ordenadores IBM que costaban millones, los «sacerdotes» los echaron de la sala.

> Trabajar con uno de esos ordenadores IBM era frustrante. No existía nada peor que el tiempo de espera entre la importación de las tarjetas perforadas y la emisión del resultado. Si te habías equivocado con una letra al preparar la instrucción, se derrumbaba el programa y te tocaba comenzar de nuevo con el proceso. Eso iba junto con la extensión de reglas malditas que envenenaban el ambiente en el centro de computación. La mayoría de las reglas sólo se inventaron para impedir el acceso a los ordenadores a locos aficionados en computación como Samson, Kotok y Sanders. La regla más estricta era aquella que decía que nadie podía tocar o manipular el ordenador. Eso era algo que los volvía locos. (Levy 1984, p.27)

A Samson y Kotok no les satisfacía mirar los ordenadores. Querían saber como funcionaban. Empezaron entonces a seguir cursos de computación para lograr así de una vez por todas averiguar como funcionaba un ordenador. Las reglas estrictas que reinaban en la sala de los computadores que funcionaban con tarjetas perforadas y los sacerdotes que supervisaban los ordenadores hacían muy difícil que se pudiera hackear. Por eso, la subcultura de los hackers se desarrolló cuando se inventó la siguiente generación de computadoras que ya no necesitaba tarjetas perforadas. Si se quería utilizar uno de los nuevos ordenadores, ya no se tenía que superar los obstáculos burocráticos tan excesivos como los que regulaban el uso de las viejas máquinas IBM. Dado que los programas se podían iniciar directamente a través de un teclado y una pantalla sin tener que importar un montón de tarjetas perforadas, los nuevos ordenadores inspiraron a los programadores a crear una nueva forma de programación y los hackers fueron sus pioneros. Además, se conectaban varios teclados y pantallas a un ordenador, lo cual hizo posible que varias personas pudieran usar un ordenador al mismo tiempo mientras se repartían las capacidades del mismo.

La ética hacker

Los jóvenes hackers desarrollaron su ética luchando diariamente por adquirir tiempo de computación y en contra de las autoridades que les impedían programar.. Aún eran pocos y todavía no tomaban muy en serio esta ética hacker, como sí llegarían a hacerlo posteriormente. La ética hacker no se publicó como manifiesto, se divulgó de boca en boca durante los primeros años. Tampoco fue discutida pero los hackers que la aceptaron la consideraron como axioma. Los puntos más importantes de la ética hacker son el acceso ilimitado a las capacidades del computador y el libre acceso de toda información. Además, hay que desconfiar de las autoridades y hay que fomentar la descentralización (Levy 1984, p. 40 y sgtes.).

Fue sobre todo la administración universitaria quien impidió fuertemente el acceso a los ordenadores para los hackers. Los sistemas abiertos sin burocracia ni autoridades les permitieron trabajar más productivamente con los ordenadores. Sin el control autoritario de los sacerdotes de IBM, lograban mayores avances con los ordenadores. Al usar las máquinas IBM, adquirían mayor poder sobre ellas. Por esta razón, es evidente que desconfiaran de cualquier otro poder que los limitara y que les impidiera usar su poder sobre los ordenadores.

Levy relata en su libro *Hackers…* la historia de la cultura hacker y de la ética hacker en el MIT hasta su fin provisional en 1984. Un capítulo entero trata de Richard Stallman, al cual Levy denomina el último verdadero hacker. Es ahí donde dice Stallman que la cultura hacker en el MIT fue un ejemplo vivo de una organización anarquista y maravillosa, antes que fuese disuelta. Stallman creó una nueva comunidad teniendo como modelo la cultura hacker, el proyecto GNU, uno de los pilares más importante del movimiento de software libre.

El movimiento de software libre abarca mucho más que el proyecto GNU. La distinción más notoria es entre los seguidores de la licencia BSD y de la licencia GPL. La licencia BSD (Berkeley Software Distribution) representa una versión del sistema operativo Unix que se originó en la Universidad de California de Berkeley a finales de los años setenta. La licencia BSD denomina hoy una serie de derivados de Unix como el FreeBSD, NetBSD o el OpenBSD. Al contrario de la licencia GPL, la licencia BSD permite que bajo ciertas condiciones se use el código fuente para crear un software propietario. No hace mucho tiempo, sólo se necesitaba agradecer públicamente a la Universidad de California.

Al comienzo de un proyecto de software, la pregunta acerca de su publicación bajo la licencia GPL o la de BSD puede ser un objeto de disputa. Si se decide usar la licencia BSD, se permite entonces a todos crear versiones comerciales basándose en su trabajo. El periodista Peter Wayner menciona en su libro *Free for All: How Linux and the Free Software Movement Undercut the High-Tech Titans* esa discusión.

> Una persona que decida sumarse al movmiento de la licencia GPL tendrá probablemente menos problemas con Richard Stallman o por lo menos, renunciará a hablar públicamente mal de él. Los seguidores de GPL tienden hacia una iconocrasia individualista, están demasiados orgullosos de sus proyectos y son impulsados por una extraña combinación de convicción personal e histeria que proviene de un autoestima demesurado. Sin embargo, los seguidores de BSD parecen ante todo más pragmáticos, más organizados y más concentrados. (Wayner 2001, 135).

Los partidarios de la licencia BSD no crean ningún culto alrededor de su licencia. En la mayoría de los casos, simplemente recalcan las libertades que proporciona la licencia BSD en comparación a la licencia GPL. Carecen de representantes carismáticos como Richard Stallmann o Linus Torvalds. Por esta razón, los proyectos BSD son ignorados por la prensa ya que no se interesan por la cruzada de Stallman para liberar el código fuente.

Sin embargo, el punto de vista de Stallman es más radical. Él propone un sistema de software libre que se parezca a Unix pero mejorado. Por esto, denominó su obra GNU que es un acrónimo recursivo para «*GNU's Not Unix*». Desde ese entonces, la meta del proyecto GNU es crear un sistema operativo completo, libre y óptimo. Desde el principio, quería ser algo más que un receptáculo de software libre. GNU es un sistema de software que quiere sustituir todo software propietario por un software GNU. Con la fundación del proyecto GNU empieza la cruzada de Stallman. La libertad que antes sólo estaba codificada en la ética hacker es expresada por un contrato legal entre el autor y el usuario a través de la licencia GNU General Public License. La licencia GPL quiere impedir que el software libre sea mal explotado.

El anarquismo de la iniciativa Open-Source

El 15 de Mayo de 1969, unidades policiales armadas bajo orden del gobernador de California Ronald Reagan irrumpen en el People's Park cerca del campo de la Universidad de California para disolver una manifestación de hippies. Una persona murió y cientos de ellas fueron heridas. La élite conservadora con su representante Ronald Reagan y la contracultura de los hippies parecían ser dos contradicciones antagónicas. No obstante, David DeLeon revela en su libro *The American as Anarchist* que el gobernador Reagan y los hippies serían más bien dos extremos del anarquismo norteamericano. Para DeLeon, el anarquismo en los EE.UU. representa una crítica radical de derecha y de izquierda hacia la sociedad liberal norteamericana. El denomina las dos alas como los «libertarios de derecha y de izquierda» (*libertarians*). Bajo este concepto, la palabra libertario sólo es una palabra diferente para decir anarquista.

Si aplicamos esta teoría de DeLeon a los partidarios de GPL y los militantes de la definición de Open-Source, nos encontramos de igual modo frente a dos extremos del anarquismo norteamericano. En su ensayo «The Cathedral and the Bazaar», Eric S. Raymond caracteriza estos dos extremos con el modelo «Catedral» y el modelo «Bazar» (ver Raymond 2001). En su libro de 1975 *The Mythical Man-Month: Essays on software engineering*, Frederick P. Brooks estableció la teoría que un proyecto de software se retrasa proporcionalmente al número de programadores que participan en el proyecto (ver Brooks 1995). Como muchos otros hackers, Eric Raymond, antiguo miembro del proyecto GNU, creía que demasiados programadores complicarían la

programación, basándose en la ley de Brook que dice que un proyecto software es más eficiente con un número pequeño de programadores. Los proyectos de software del proyecto GNU se componen también de pocos programadores. Bajo el asombro de Raymond, Linus Torvalds demostró lo contrario publicando rápidamente su proyecto Linux: Mientras más hackers invitó a trabajar en el proyecto Linux, mejor resultaba.

Raymond anotó sus observaciones en el ensayo «The Cathedral and the Bazaar», donde pone en contraposición los diferentes estilos de dirección del proyecto GNU y del proyecto Linux. Desarrolló este ensayo a partir de una conferencia que dio por primera vez en 1997 en un congreso de Linux en Alemania. El título del ensayo proviene de su analogía central: los programas GNU serían más bien catedrales impresionantes, monumentos de la ética hacker planeadas centralmente, creados para la eternidad. En cambio, el proyecto Linux aparenta ser más un bazar lleno de comerciantes chismosos. Dentro de esta analogía, también se encuentra una comparación entre Stallman y Torvalds. En este caso, Stallman sería un representante clásico de los arquitectos de catedrales. Es el gurú de la programación que puede desaparecer por 18 meses para reaparecer con un genial compilador de C. Al contrario, Torvalds sería el anfitrión de una fiesta, distribuye las discusiones sobre el diseño de Linux entre los grupos del proyecto. Sólo interviene si hay tanta discordia en un grupo que se necesita un arbitro. A fin de cuentas, es Torvalds el que decide lo que hará parte del *Kernel*. Su función más importante consiste en mantener el flujo de ideas.

El análisis de Raymond le dio la fama de ser un «evangelista del mercado libre». Él está contra la intervención del estado en el mercado. Para él, el individuo debe ser reforzado a través de la desregularización, lo que también incluye el derecho de portar armas. DeLeon le asignaría en *The American as Anarchist* en la categoría de «libertario derechista». Los libertarios derechistas son aquellos que opinan que el gobierno debe dejarlos tranquilos para que ellos hagan lo que les apetezca con su dinero y sus armas. En el momento en que Raymond analizó el movimiento de software libre y su anarquismo productivo, descubrió lo que como libertario derechista quería descubrir: un mercado libre no reglamentado. Según él, la base del éxito del movimiento de software libre es la libertad del usuario. El modelo «bazar» ya representa en sí una máxima desregularización y libertad ya que diversos comerciantes compiten entre ellos. Al contrario, los proyectos GNU o la creación de software propietario por parte de empresas se estructuran de forma parecida a las

comunidades medievales: la construcción de la catedral es empujada con el dinero de la ciudad por parte de un grupo de curas y representa la realización de las ideas de un arquitecto. Bajo estas condiciones, la obra sólo se puede realizar si hay suficiente dinero, un arquitecto bien dotado y trabajadores. Al contrario, en el «bazar» los diferentes comerciantes intentan arruinarse mutuamente. El ganador de esta competencia tiene entonces la mayoría de los clientes, visto bajo un concepto social-darwinista: él que mejor se adapta es el que sobrevive.

El problema del ensayo de Raymond es que ni los proyectos de GNU son catedrales puras, ni el proyecto Linux es realmente un «bazar», especialmente el proyecto Linux con sus publicaciones a corto plazo y sus miles de colaboradores, Linus Torvalds sigue estando en la punta de la pirámide, es él que decide finalmente lo que podrá formar parte del Kernel. El proyecto Linux se puede definir mejor como una forma combinada, ni totalmente una catedral, ni totalmente un bazar.

En su conferencia de 1997, Raymond tambien habló del software libre. En febrero de 1998, reemplazó el término software libre por el de Open Source. Para él y algunos de los partidarios del movimiento de software libre, Stallman se convirtió en un estorbo. Pensaban que Stallman daba una mala imagen a los comerciantes con sus comentarios políticos que parecían ser muy comunistas. Además, querían impedir que el movimiento se concentrara demasiado en la licencia GPL. Deseaban un sistema de software que diera espacio no sólo al software de licencia GPL, sino también a otras licencias como BSD o similares. Y este sistema lo llamaron Open Source. En su libro *Freie Software — Zwischen Privat- und Gemeineigentum* Volker Grassmuk comenta el hecho de la siguente manera:

> «*Free*» no sólo tiene un doble sentido («cerveza gratis» o «libre exprexión»), sino se había transformado obviamente en «*The land of the Free*», en una palabra indecente, de confrontación, de alguna forma en la palabra de cuatro letras con conotación comunista. (Grassmuck 2002, p.230)

El último verdadero hacker

En su libro *Hackers*, Levy entrevistó también a Richard Stallman. Dedica un capitulo entero a su vida, que no sin razón tituló «*Epilogue: The Last Of The True Hackers*» (Epílogo: el último de los hackers), puesto que en 1984 la situación era pésima para el software libre. Stallman pertenece a la generación de los primeros hackers que

habían programado máquinas gigantescas IBM en las universidades norteamericanas. La gente joven que empezó a aparecer en los años ochenta en las salas de computación de las universidades aprendían a programar en su ordenador personal, libre de cualquier ética o comunidad hacker.

> Estos jóvenes programaban tan bien como sus antecesores. Pero con ellos apareció un fenómeno nuevo que consistía en el hecho de que sus programas aparecían en las pantallas con Copyrights. Para Stallman, quien defendía la idea de que toda información debe fluir libremente, este hecho era una blasfemia. «Yo pienso que el software no debe pertenecerle a nadie» ya que esta práctica sabotea a toda la humanidad. Eso impide que la gente saque el máximo provecho del software. (Levy 1984, p.419)

Estos nuevos hackers no se interesaron mucho en la ética hacker. Stallman había aprendido en el centro de computación del MIT que una institución anarquista podía ser posible. Lo único que le faltaba era más gente que participara. La razón de este fenómeno era la descentralización por el hecho que ahora, cada hacker tenía su propio ordenador en casa. A principios del los años ochenta, él se sentía como el último partidario de un movimiento muerto organizado en torno a los principios anarquistas de la ética hacker. Quería reanimar este movimiento. Con el movimiento de software libre renace la cultura hacker y Stallman se propone liberar el código fuente de toda licencia propietaria.

El movimiento de software libre en la forma de GNU, BSD o iniciativa Open Source es la crítica radical y anarquista al orden de la propiedad intelectual no sólo en la sociedad liberal de EE.UU., sino en el mundo entero globalizado. Al contrario de los representantes de la licencia BSD o del anarquismo de libre mercado de Eric Raymond, Stallman aboga por un anarquismo cooperativo que bajo las palabras del anarquista francés Jean-Paul Proudhon plantea que la propiedad es un robo. Para la mayoría, la abolición de la propiedad intelectual es impensable. No obstante, hace medio milenio, la propiedad privada era inconcebible. Así lo comenta Jeremy Rifkin en su libro *Access*:

> Que superemos el sistema de mercado y el intercambio de mercancías es una cosa inimaginable para la mayoría de la gente, como había sido algo inconcebible hace medio milenio cercar y privatizar el terreno y el trabajo y de esa forma integrarlos en relaciones de propiedad privada. (Rifkin 2000, p.14)

Stallman y la fracción de GNU del movimiento de libre software quieren liberar no sólo la propiedad intelectual en forma de software de las licencias propietarias, sino también los libros y la música. Lo declara Stallman en una conversación publicada en Spiegel Online: «Yo voy más en dirección a la idea anarquista de izquierda, que nos juntemos libremente y reflexionemos cómo podemos mantener a todos cooperando.» (Klagges 1996).

Traducción de Vicente Vargas Martínez

Bibliografía

Barbrook, R. und Cameron, A. (1995), `The Californian Ideology‘. *http://www.alamut.com/subj/ideologies/pessimism/califIdeo_I.html* [30th November 2004].

Brooks, F. P. (1995), *The Mystical Man-Month: Essays on software engineering*. New York.

DeLeon, D. (1978), *The American as Anarchist*. Baltimore, MA.

Grassmuck, V. (2002), *Freie Software – Zwischen Privat- und Gemeineigentum*. Bonn.

Imhorst, C. (2004), *Die Anarchie der Hacker – Richard Stallman und die Freie-Software-Bewegung*. Marburg.

Klagges, H. (1996), `Es reicht mir nicht, nur einfach neugierig auf die Zukunft zu sein, ich will etwas ändern‘. Interview with *Richard Stallman. http://www.klagges.com/pdf/interview_stallman.pdf* [30th November 2004].

Levy, S. (1984), *Hackers: Heroes of the Computer Revolution*. New York.

Raymond, E. S. (2001), *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA.

Rifkin, J. (2000), *The Age Of Access*. New York.

Wayner, P. (2001), *Free For All. How Linux and the Free Software Movement Undercut the High-Tech Titans.* New York.

Williams, S. (2002), *Free as in Freedom – Richard Stallman's Crusade for Free Software*. Sebastopol, CA.

Manifiesto de la Guerrilla por el Acceso Abierto

La información es poder. Pero como todo poder, hay quienes quieren quedarse con él. Todo el patrimonio cultural y científico del mundo, publicado durante siglos en libros y diarios, continúa siendo digitalizado y guardado por un puñado de corporaciones privadas. ¿Quieres leer publicaciones acerca de los más famosos resultados de la ciencia? Necesitarás enviar grandes cantidades a editoriales como Reed Elsevier.

Existen personas luchando para cambiar esto. El Movimiento para el Acceso Abierto ha luchado valientemente para asegurar que los científicos no firmen derechos de autor y en cambio se aseguren que su trabajo sea publicado en Internet, bajo términos que permitan que cualquier persona tenga acceso a este. Pero incluso en el mejor de los casos, su lucha solamente aplicará para cosas que se publiquen en el futuro. El resto, lo publicado hasta ahora, se habrá perdido.

Este es un precio demasiado alto para pagar. ¿Obligar a que académicos paguen dinero para leer el trabajo de sus colegas? ¿Digitalizar bibliotecas enteras pero solo permitir que la gente en Google las pueda leer? ¿Proveer artículos científicos a aquellos en las élites universitarias del primer mundo, pero no a niños en el sur del planeta? Es indignante e inaceptable.

«Estoy de acuerdo», muchos dicen, «pero ¿qué podemos hacer?. Las compañías mantienen los derechos de autor, ganan enormes cantidades de dinero al cobrar por el acceso, y todo es perfectamente legal. No hay nada que podamos hacer para detenerlas». Pero sí hay algo que podemos hacer, algo que ya se está haciendo: podemos contraatacar.

Quienes tienen acceso a estos recursos —estudiantes, bibliotecarios, científicos— han recibido un privilegio. Pueden alimentarse de este banquete de conocimiento mientras el resto del mundo es excluido. Pero ustedes no necesitan —de hecho, moralmente, no pueden— mantener este privilegio solamente para ustedes. Tienen el deber de compartirlo con el mundo. Y tienen que compartir claves con sus colegas y llenar solicitudes de descargas para sus amigos.

Mientras tanto, aquellos que han sido excluidos no esperan sin hacer nada. Han estado fisgoneando a través de agujeros y trepando cercas, liberando información guardada por las editoriales y compartiéndola con sus amigos.

Pero todas estas acciones se quedan en la oscuridad, escondidas en el sótano. Se las llama robo o piratería, como si compartir la riqueza del conocimiento fuese el equivalente moral a saquear un barco y asesinar a su tripulación. Compartir no es inmoral: es un imperativo moral. Solo quienes están cegados por la ambición podrían rehusarse a dejar que un amigo obtenga una copia.

Las grandes corporaciones, por supuesto, están cegadas por la ambición. Las leyes bajo las cuales operan lo requieren. Sus accionistas se sublevarían si fuese menos que esto. Y los políticos que han comprado las respaldan, aprobando leyes que les conceden el poder exclusivo para decidir quiénes pueden hacer copias.

No hay justicia al cumplir leyes injustas. Es hora de salir a la luz y, siguiendo la tradición de la desobediencia civil, oponernos a este robo privado de la cultura pública.

Necesitamos tomar la información, donde quiera que esté almacenada, hacer copias y compartirlas con el mundo. Necesitamos tomar cosas que ya no tienen derechos de autor y agregarlas al archivo. Necesitamos comprar bases de datos secretas y publicarlas en la web. Necesitamos descargar publicaciones científicas y subirlas a redes de intercambio de archivos. Necesitamos combatir en la Guerrilla del Acceso Abierto.

Con suficientes de nosotros, alrededor del mundo, no solo enviaremos un mensaje firme en contra de la privatización del conocimiento. Haremos que sea una cosa del pasado.

¿Te unirás a nosotros?

Aaron Swartz

Julio de 2008, Eremo, Italia

No más móviles a partir de ahora

Traducción del inglés por Jorge Maldonado Ventura del artículo de Alyssa Rosenzweig «No Cellphones Beyond This Point»[10] bajo la licencia CC by-SA

Me niego a llevar un móvil —para mis confusos amigos en nuestra sociedad obsesionada con la tecnología—, aquí explico por qué. Algunos de vosotros ya me habéis preguntado cuál es mi número para enviarme mensajes. Quizá eras un profesor en una de mis clases pidiéndome que ejecutara algún software privativo en clase. Puede que fueras un familiar, preocupado de que en una situación insegura no fuera capaz de llamar para pedir ayuda.

Hay cuatro capas de razonamiento detrás de mi negativa a tener un celular, a pesar de ser una usuaria activa de Internet. En orden de menor a mayor importancia:

Primero, la electrónica de los móviles es incómoda para mí. Gran parte de mi tiempo frente al ordenador lo empleo en la escritura, la programación y el arte; para mí estas tareas requieren teclados de tamaño completo o tabletas de dibujo. Esta no es una razón ética para evitar teléfonos y tabletas, por supuesto, y reconozco que muchas personas tienen usos más adecuados para los factores de forma pequeños.

Segundo, los usuarios de celulares crean una cultura del celular. En una fracción de la vida de un adulto, los celulares han cambiado desde la inexistencia hasta usarse de forma socialmente aceptable mientras se habla con alguien en la vida real. Esta cultura no es inevitable para la electrónica digital —mucha gente usa la tecnología responsablemente, por lo que les aplaudo—, pero permanece depresivamente usual. Si tuviera que tener un teléfono frente a mi nariz mientras finjo hablar con mis propios amigos, solo seguiría perpetuando la noción de que *es un comportamiento correcto*. Como temo que pueda convertirme en alguien que usa inadecuadamente la tecnología de esta manera, evito llevar un celular completamente para evitar el riesgo ético.

Tercero, los celulares suponen riesgos graves para la libertad y la privacidad. La amplia mayoría de los teléfonos del mercado ejecutan sistemas operativos privativos, como iOS, y están repletos de software privativo. Asimismo, a diferencia de la mayoría de ordenadores portátiles y de escritorio, muchos de estos sistemas operativos ejecutan comprobaciones de firma. Es decir, es criptográficamente imposible y en muchos casos ilegal reemplazar el sistema por software libre. Esto por sí solo es una razón para evitar tocar estos aparatos.

10 https://rosenzweig.io/blog/no-cellphones.es.html

La situación real es desafortunadamente peor. En la electrónica convencional hay un único chip principal dentro, la UCP [Unidad Central de Procesamiento]. La UCP ejecuta el sistema operativo, como GNU/Linux[11], y tiene el control total de la máquina. No es de esta forma para celulares; estos aparatos tienen dos chips principales —la UCP y la banda base—. La primera tiene el conjunto habitual de problemas de libertad; la última es una caja negra conectada a través de Internet con un conjunto de capacidades espeluznante. Como mínimo, debido al diseño de las redes telefónicas, en cualquier momento en que el teléfono está conectado a la red (es decir, la banda base está en línea), la ubicación del usuario puede ser rastreada triangulando la señal de las torres telefónicas. El riesgo es ya inaceptable para muchas personas. Las operaciones de telefonía tradicionales son vulnerables a la vigilancia y a la manipulación, al no ser ni las llamadas ni los mensajes encriptados. Y, para colmo, pocos teléfonos proporcionan un aislamiento aceptable de la banda base. Es decir, la UCP, que podría ejecutar software libre, no controla la banda base, lo cual a efectos prácticos hace ilegal ejecutar software libre en los Estados Unidos. Más bien, en muchos casos, la banda base controla la UCP. No importa si se usa mensajería encriptada sobre XMPP si la banda base puede simplemente tomar una captura sin el conocimiento ni el consentimiento del sistema operativo del lado de la UCP. Alternativamente, de nuevo dependiendo de cómo esté la banda base conectada al resto del sistema, puede tener la capacidad de activar remotamente el micrófono y la cámara. 33 años después, un mundo en el que cada persona lleva un celular supera las pesadillas de George Orwell. Puede que no tengas «nada que esconder», pero yo al menos todavía me preocupo por mi privacidad. Los celulares son espeluznantes. No contéis conmigo.

Finalmente, ante las graves implicaciones para la sociedad y la libertad, me niego a perpetuar este sistema. Podría decidir llevar un celular de todas formas, decidiendo que como una persona aburrida puedo sacrificar la libertad en nombre de la instantáneamente gratificante conveniencia. Pero al ser complaciente, solamente aumentaría el tamaño del problema, una pesada carga ética al usar el celular contribuye al efecto de red[12], como el nombre sugiere.

Si tuviera mi teléfono en frente de otros, estaría mostrando que «los celulares están bien». Si alguien me tiene como ejemplo ético, puede que ese alguien siguiera usando un celular.

11 https://gnu.org
12 https://es.wikipedia.org/wiki/Efecto_de_red

Si dejara que mis amigos me enviaran mensajes de texto en vez de usar medios más éticos, estaría señalando que «enviar mensajes de texto está bien» y «es razonable esperar que las personas se envíen mensajes de texto». Si permanecieran pasivos ante la ética y necesitaran llevar un teléfono, esto podría empujarles a preservarlo.

Si usara un teléfono para actividades en clase, estaría indicando que «los estudiantes del siglo 21 deberían tener un teléfono». Preferiría ser la última resistencia en la clase para recordarles que esta no es una asunción ética.

Si recibo una mirada desconcertada de mis conocidos, confidentes y profesores, ahora tengo la oportunidad de educarles acerca del software libre y la privacidad. Pocas personas son conscientes de los peligros de estos «aparatos de vigilancia portables» como escribiría Richard Stallman[13]. Estos «momentos extraños» son oportunidades perfectas para ayudarles a tomar una decisión más informada.

Al llevar un celular estaría perpetuando algo malvado. Al negarme a llevar uno, opongo resistencia y hago algo bueno activamente.

Bueno, ¿en vez de usar un celular, cuáles son mis alternativas?

Para la mayoría de tareas digitales, incluida la escritura de esta publicación, uso un portátil que ejecuta software libre[14]. Además, ¡para conectarme a Internet, uso una tarjeta de Wi-Fi[15] que ejecuta firmware libre!

Para hablar con mis amigos, uso protocolos descentralizados y de especificaciones libres cuando es posible. En concreto, estoy disponible en correo electrónico, XMPP y Mastodon. En algunos casos en los que esto no es posible debido al efecto de red, uso sistemas centralizados libres como IRC. En ocasiones, uso sistemas privativos a los que se les ha aplicado la ingeniería inversa para su uso con software libre, como Discord[16]. De ser posible, me cubro de capas de encriptación implementadas con software libre, como GPG y OTR, para una protección extra contra amenazas de privacidad. Si la privacidad de la ubicación es un problema, me conectaré a través de Tor[17]. Cualquiera de estas medidas es un paso importante más allá de las llamadas de teléfono, mensajería de texto, WhatsApp o Snapchat. Todas ellas juntas te protegeran de la mayoría de adversarios.

13 https://stallman.org/
14 https://es.wikipedia.org/wiki/Software_libre
15 https://es.wikipedia.org/wiki/Qualcomm_Atheros
16 https://rosenzweig.io/git/?p=purple-discord.git
17 https://www.torproject.org

Para conectarme cuando estoy lejos de casa, busco redes Wi-Fi públicas, que pueden hacerse seguras con encriptación y con Tor. Si esto no es una opción, puede que necesite preguntar alguien para que me preste sus electrónicos —esto es desgraciado, pero mientras que el efecto de red está en juego, es éticamente aceptable explotarlo —. La mayoría del tiempo evitaré conectarme a Internet fuera de casa de todas formas; ¡soy más productiva desconectada!

Así que, sí, puedo vivir sin un celular. No es siempre conveniente, pero la productividad, la libertad y el comportamiento ético siempre prevalecen sobre la conveniencia.

Os animo a hacer lo mismo.

Definición de cultura libre[18]

Para que una obra sea considerada libre, *debe* estar cubierta por una licencia cultural libre, o su situación legal *debe* proporcionar las mismas *libertades esenciales* enumeradas anteriormente. Sin embargo, no es una condición suficiente. De hecho, una obra específica puede ser no-libre de otras maneras que restrinjan las libertades esenciales. Éstas son las condiciones adicionales para que una obra sea considerada libre:

- **Disponibilidad de los datos fuente:** Si un trabajo final ha sido obtenido mediante la compilación o procesamiento de uno o varios ficheros fuente, todos los datos fuente subyacentes deben estar disponibles junto con el propio trabajo bajo las mismas condiciones. Esto puede ser la partitura de una composición musical, los modelos usados en una escena 3D, los datos de una publicación científica, el código fuente de una aplicación informática, o cualquier otra información similar.
- **Uso de un formato libre:** Para los ficheros digitales, el formato en que se haga disponible el trabajo no debe estar protegido por patentes, salvo que se conceda un permiso libre de regalías, ilimitado, irrevocable y de ámbito mundial para hacer uso de la tecnología patentada. Si bien en ocasiones se pueden usar

18 https://freedomdefined.org/Definition/Es

formatos no-libres por razones prácticas, *debe* haber disponible una copia en un formato libre para que el trabajo sea considerado libre.

- **Sin restricciones técnicas:** La obra debe estar disponible de una forma en la que no se usen medidas técnicas para limitar las libertadas enumeradas anteriormente.
- **Sin otras restricciones o limitaciones:** La propia obra no debe estar cubierta por restricciones legales (patentes, contratos, etc.) o limitaciones (como derechos de privacidad) que impidan las libertades enumeradas anteriormente. Una obra puede hacer uso de las existentes exenciones legales al copyright (para citar obras protegidas), pero sólo las partes de ella que sean libres sin ambigüedades constituyen una obra libre.

En otras palabras, siempre que el usuario de una obra no pueda ejercitar legalmente o en la práctica sus libertades básicas, la obra no puede ser considerada ni llamarse «libre».